KGCOWORKING у сарадњи са

презенц

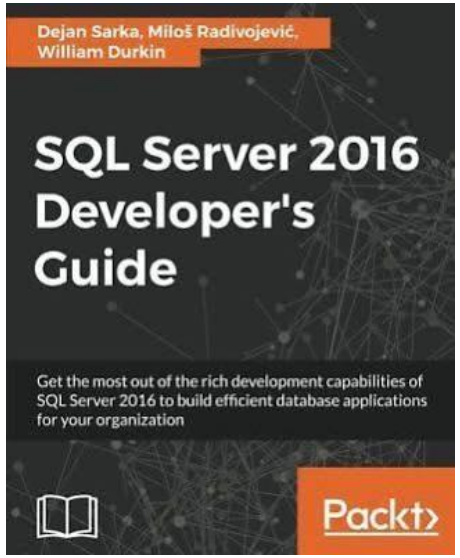
ДЕЈТАБЕЈЗ ДИЗАЈН
МИТОВИ И ТИПИЧНЕ ГРЕШКЕ

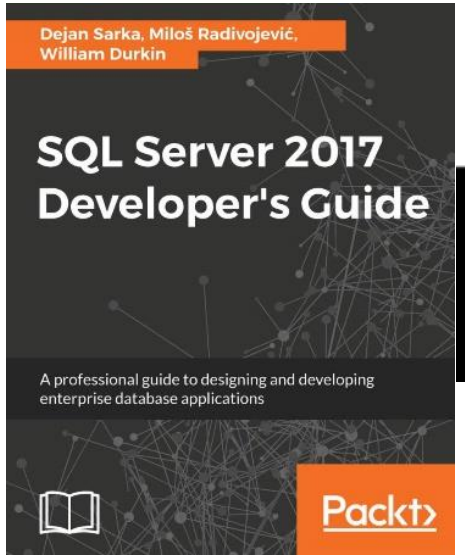
МИЛОШ РАДИВОЈЕВИЋ, BWIN, АУСТРИЈА

КРАГУЈЕВАЦ, 18. АПРИЛ 2019.

Милош Радивојевић

- Data Platform MVP
- Principal Database Consultant, bwin GVC Vienna, Austria
- Co-Founder: SQL Pass Austria
- Conference Speaker, Book Co-Author

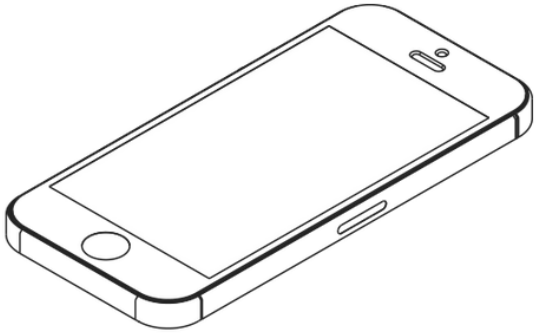




Contact:
MRadivojevic@gvcgroup.com
www.bwinparty.com
Twitter: **@MilosSQL**

Упитник

Go to **www.menti.com** and use the code **22 60 48**



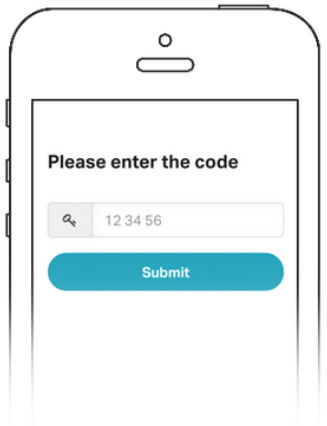
1

Grab your phone

www.menti.com

2

Go to www.menti.com



3

Enter the code 22 60 48 and vote!

Зашто је битан добар дизајн?

- Коко год звучало патетично, то је заиста темељ пројекта
- Промене су веома скупе и у касним фазама развоја софтвера практично немогуће
- Добар дизајн чува паре, време и репутацију
- Барем три дб дизајна треба да буду имплементирана:
  - Концептуални, логички и физички дизајн

Проблеми са дизајном база података

Данас сви играју фудбал <<<
Данас сви раде у ИТ-у

- Где се учи дејтабејз дизајн?
- Ко дизајнира базе и табеле?
- Недовољно присутна тема у мору дејтабејз тема
  - SQLSaturday Conference:
    - 7390 session titles
    - only 16 titles with the database design in its name

Проблеми са дизајном база података

- РДБМС готово да не подржавају алате за дизајн (SQL Server нпр.)
- Agile – брт мора да завршимо до среде, немамо кад да разматрамо сва та питања, после ћемо
- Људи не архивирају податке, све рокају у главну базу
  - база расте
  - Индексираш старе податке
  - Читаш више него што ти треба
  - У меморији буду и подаци који ти не требају

- Користан линк: http://www.databaseanswers.org/

Митови у вези са дизајном база података

Credits to Dejan Sarka
dsarka@solidq.com, dsarka@siol.net, @DejanSarka

Storage Myth

- **Relational database is just a storage!**

- **WRONG!**
  - Relational model is about data integrity
  - RDBMS enforces data integrity

Physical Myth

- **Knowing details of a RDBMS is very important**
  - **You should design for performance, not for some relational theory!**

- **WRONG!**
- Logical problems are even more important than physical
  - Physical problems pass over
  - Relational model is a logical model
  - A good logical design should perform well; the opposite is not always true

Repeating Group Myth

- **A table with repeating group of columns needs normalization (1st NF)**

```sql
CREATE TABLE dbo.Employees(
EmpId INT NOT NULL PRIMARY KEY,
EmpName NVARCHAR(50),
Child1Name NVARCHAR(50),
Child2Name NVARCHAR(50),
Child3Name NVARCHAR(50)
```

- **WRONG!** Let's rename the table and the columns

```sql
CREATE TABLE dbo.Employees(
EmpId INT,
EmpName NVARCHAR(50),
Ocuppation NVARCHAR(50),
FavoriteSportClub NVARCHAR(50),
PrimarySchoolName NVARCHAR(50)
```

Repeating Group Myth

- Repeating columns with similar names is not a repeating group!
- Just a constraint built in the model
- This is normalized:

```
ORDERS(OrderId, CustomerId, OrderDate, DueDate,
ShipDate,...)
```

- And this not?

```
EMPLOYEES(EmpId, EmpName, Child1Name, Child2Name,
Child3Name,...)
```

Atomicity Myth

- **Every collection in a single column of row is a repeating group, is not atomic**

- **WRONG!**

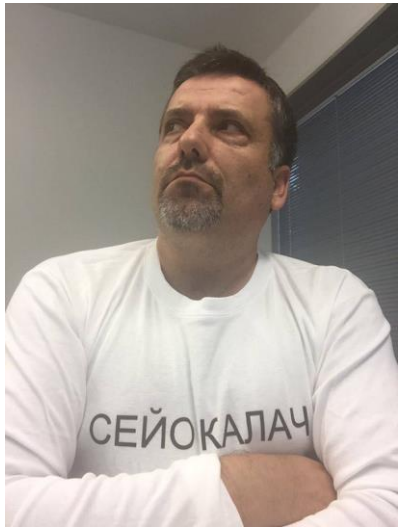
- We need to know the correct definition of atomicity
  - A value is atomic as long you operate with it only with methods defined on the data type of value (e.g. DATETIME data)

Think Set Oriented - Running Total

- **What does set oriented mean for running total?**
  - **Selfjoin**
  - **Cursor solution**
  - **Window aggregate function**
- **Which solution is set oriented?**
  - **Selfjoin**
- **Which solution is the slowest one and not scalable?**
  - **Selfjoin**

- DEMO

DB Design Common Mistakes



Агенда

- Do Not Create Table in Two Minutes!
- Одабир одговарајућег типа података
- Коришћење ограничења (констрејнта)
- Редуковање колона које прихватају НУЛЛ
- Спољни кључеви
- Имена објеката (Naming conventions)
- Куде припада бизнис логика?
- Природни кључеви ака спаваш ли мирно, Џоне Селко?

Do Not Create Table in Two Minutes!

- Како програмер обично прави табелу?
- Нађе CREATE TABLE неке табеле, удари Copy, удари Paste, мало промени имена колона, изабере примарни кључ и тоето
- Do not create table with copy/paste!
- Немојте људи!

Do Not Create Table in Two Minutes!

- Имплементација пословне логике почиње са CREATE TABLE
- Одвојите довољно времена приликом анализе захтева
- Размишљајте о подацима: процена количине података, колико ће се и како мењати
- Уверити бизнис страну да је то битно, да треба времена и да је то инвестиција у скалабилност и стабилност пројекта
- Озбиљне промене и ридизајн су скупе, дуготрајне, неретко и мучне

Do Not Create Table in Two Minutes!

Unless you are

Одабир одговарајућег типа података

Ако изабереш робуснији тип података који није неопходан

- Потребно је више простора на диску на серверу и копијама, бекап је спорији...
- SQL Server мора да учита мање страница за исти резултат

Изабери DATE тип податка, ако није потребна прецизност у сатима или минутима

Промене типа података су веома скупе, тешко се продају, а углавном нико није заинтересован да за њих плати

Одабир одговарајућег типа података

Промене које су неопходне у систему када мењате тип података:

- Ажурирање типа података на нивоу табеле
- Ажурирање параметара ускладиштених процедура које користе колону чији се тип мења
- Ажурирање класе у апликационом коду
- Ажурирање апликационог интерфејса према подацима тзв. дејта аксес лејера
- Ефекти промене су тешко мерљиви и потпуно невидљиви за крајњег корисника код кога се обично налази кеса са новцима

Одабир одговарајућег типа података

```
ALTER TABLE dbo.StudentExams  ALTER COLUMN exam_note TINYINT;
```

```
IF OBJECT_ID('dbo.insertStudentExams','U') IS NULL
    DROP PROCEDURE dbo.insertStudentExams
GO
CREATE PROCEDURE dbo.insertStudentExams
@student_id int, @exam_id int, @exam_note tinyint
AS
INSERT dbo.StudentExams VALUES(@student_id,@exam_id,@exam_note)
```

```
//se.ExamNote = dr.GetInt32(2);
se.ExamNote = (int) dr.GetByte(2);
```

Коришћење ограничења (констрејнта)

Користи ограничења како због интегритета података тако и због перформанси

CHECK Constraints

- Ограничите вредности које су дозвољене у оквиру колоне
- Нпр. ако је колона Оцена
  - Онда је смислено поставити ограничење 1-5; тада нико не може да унесе ништа друго
  - Сви корисници ће интерпретирати вредности у колони на исти начин
  - Ако колона има тип података INT, без чек констрејнта, неко ће кад-тад да упише 999 и то да интерпретира по сопственом нахођењу
- Подиже квалитет података за потоњу анализу (DWH/BI)

UNIQUE Constraints

- Ако је нешто јединствено, употребите UNIQUE констрејнт и поделите информацију о јединствености са RDBMS системом

Коришћење ограничења (констрејнта)

Именовање констрејнта

Увек их именујте

```
create table dbo.k (id int not null,
c1 date default getdate()
)
go
alter table k alter column c1 datetime
```

100 %

Messages

```
Msg 5074, Level 16, State 1, Line 5
The object 'DF__k__c1__276EDEB3' is dependent on column 'c1'.
Msg 4922, Level 16, State 9, Line 5
ALTER TABLE ALTER COLUMN c1 failed because one or more objects access this column.
```

Редуковање колона које прихватају НУЛЛ

Смањити број колона које прихватају НУЛЛ

- Уноси оверхеад на нивоу сваког реда
- Апликациони код мора да хендлује сетове података који садрже НУЛЛ

Различити актори обично имају различите интерпретације редова у којима колна има NULL

Покажите да вам је стало до квалитета и конзистентности података

Некада је немогуће избећи NULL, али будите сигурниц да сте бар покушали

Ако не умете да објасните зашто колона подржава НУЛЛ онда вероватно не мора то да ради

Спољни кључеви

Не заборавите на спољне кључеве (foreign keys)

Урадите то на време, приликом прављења табеле, не остављајте за касније, касније неће да дође

Ако нисте заборавили спољне кључеве будите сигурни да сте креирали индексе над њима (уколико то има смисла)

Имена објеката (Naming conventions)

fDate

- Који је то датум: старт, кат оф, енд или трансакцион тајм стемп?
- Свашта питаш, јбт, па старт дејт!
- А је л UTC, GMT или који већ формат?
- Па UTC, брт, знаш ваљда да од 15. априла у систему X подсистем Y користимо само UTC, а GMT само у систему Z и то у верзији в3.x
- А што се колона једноставно не зове **fStartDateUTC**, па бисмо обојица били поштеђени овог испразног дијалога?!

Имена објеката (Naming conventions)

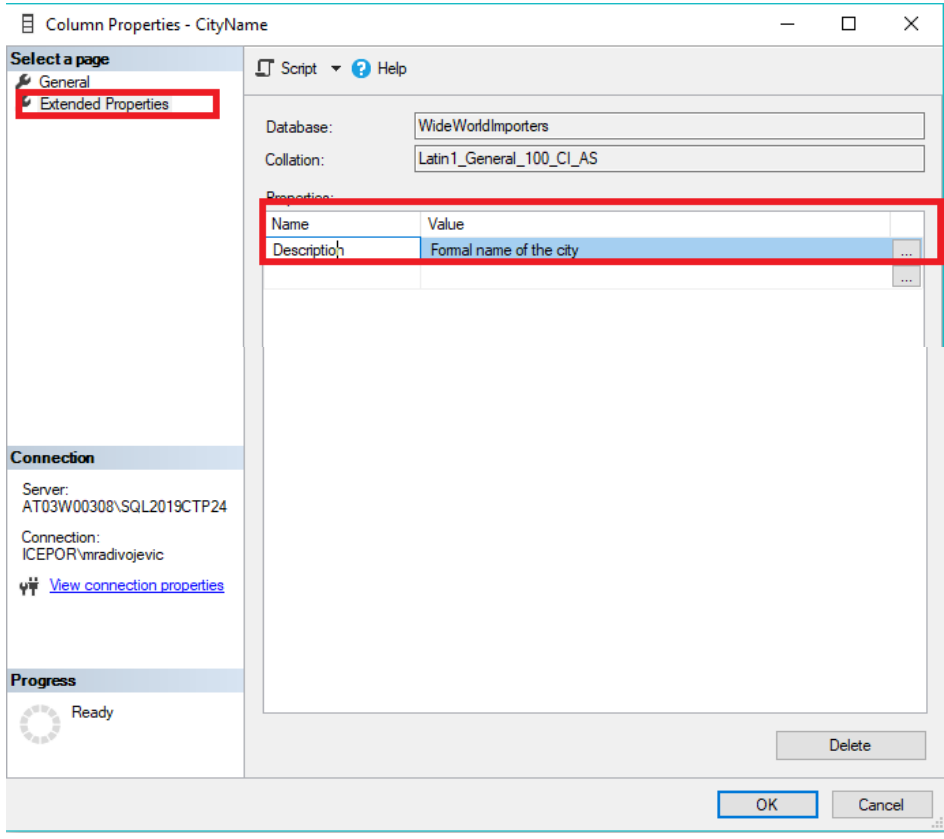
Боље икаква нејминг конвенција него никаква

Избегавати:

- Спејс карактер у имену
- Кључне речи
- Скраћенице
- Негације (**isValid** уместо **isNotValid**)

Документовати табеле и колоне:

ExtendedProperties



Where Does Business Logic Belong?

Имплементација интегритета података на нивоу апликације

- База је трајнија од апликације
- Апликације је више (поновна искористивост кода)

Пословна логика припада апликацији, база је само магацин са подацима

- Један од опаснијих митова у програмирању уопште
- Sorry pal, али SELECT стејтмент је имплементација бруталне бизнис логике, чак и CREATE TABLE уводи озбиљне рестрикције у дозвољени тип и скуп података

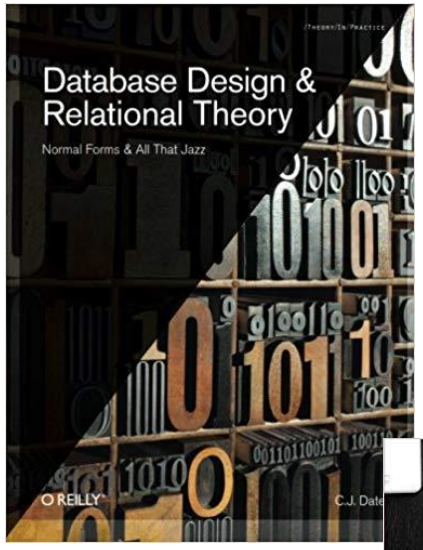
Пословна логика је тамо ђе сам ја најбољи!

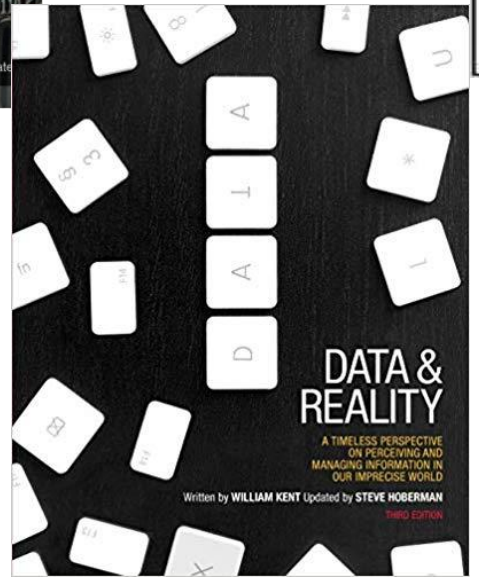
Природни кључеви илити спаваш ли мирно, Џое Селко?

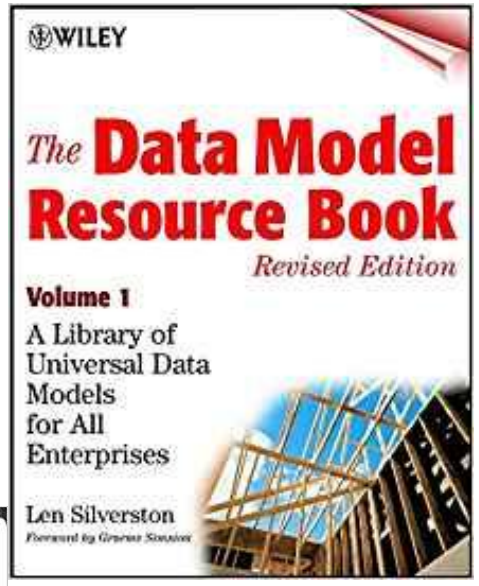
Природни кључеви и ГДПР

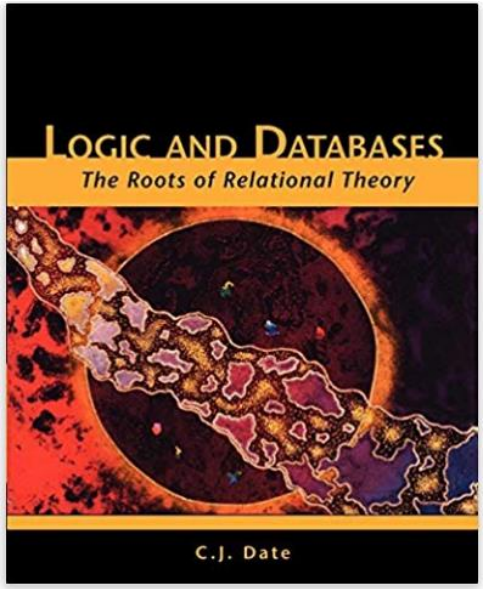
- Ажурирање табеле и свих процедура јербо сад не смеш да чуваш личне податке, ако се неки добро обавештени а намћорасти грађанин (ака трипер) побуни

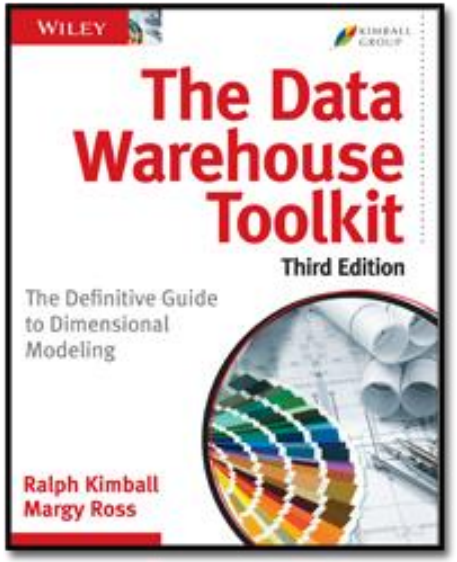
Препоручене књиге



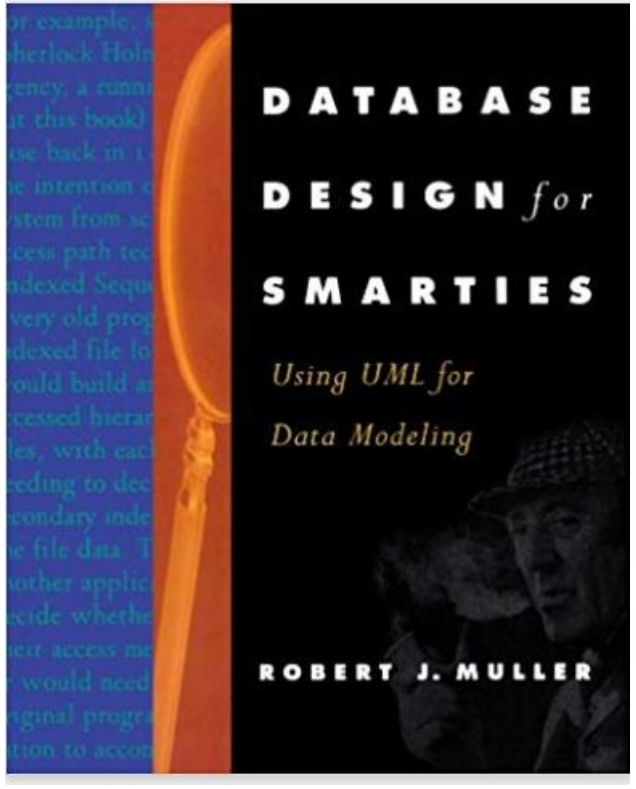


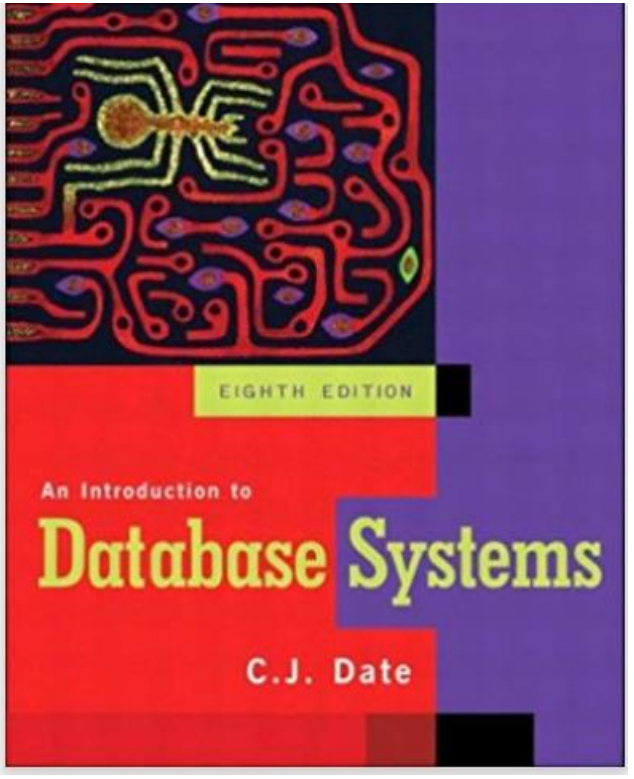






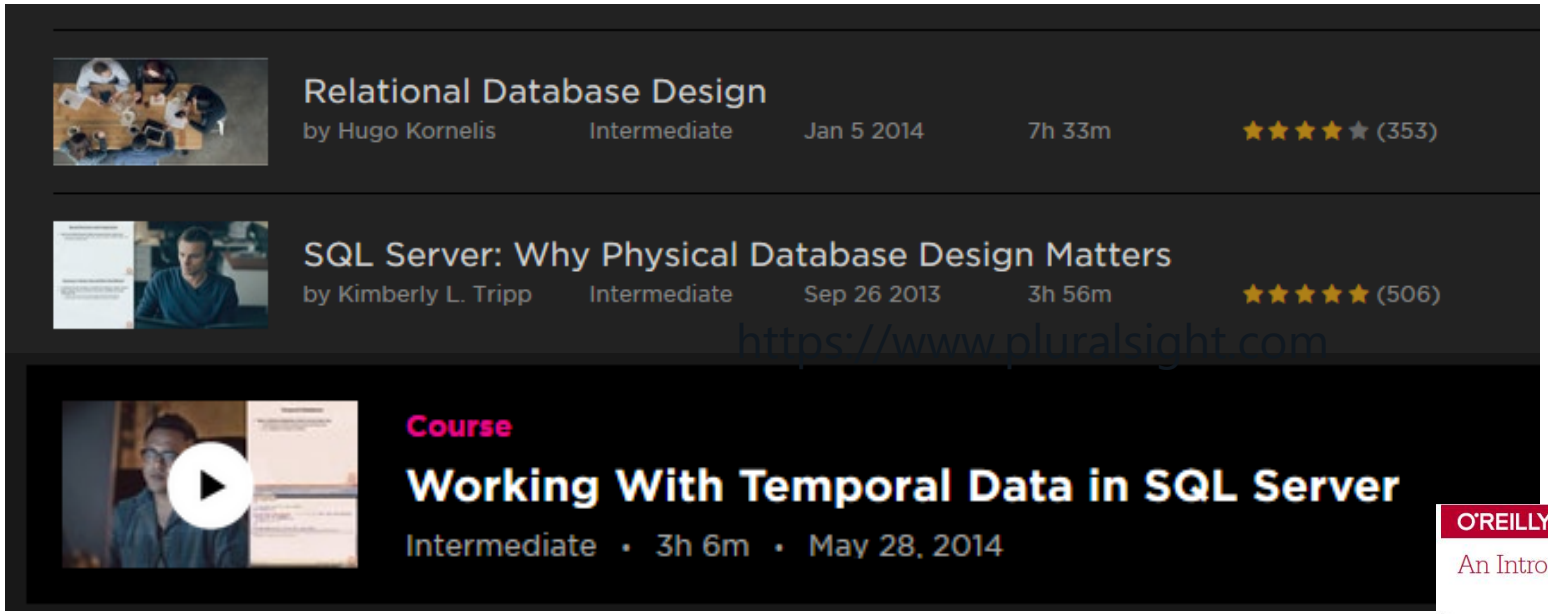
Препоручене књиге

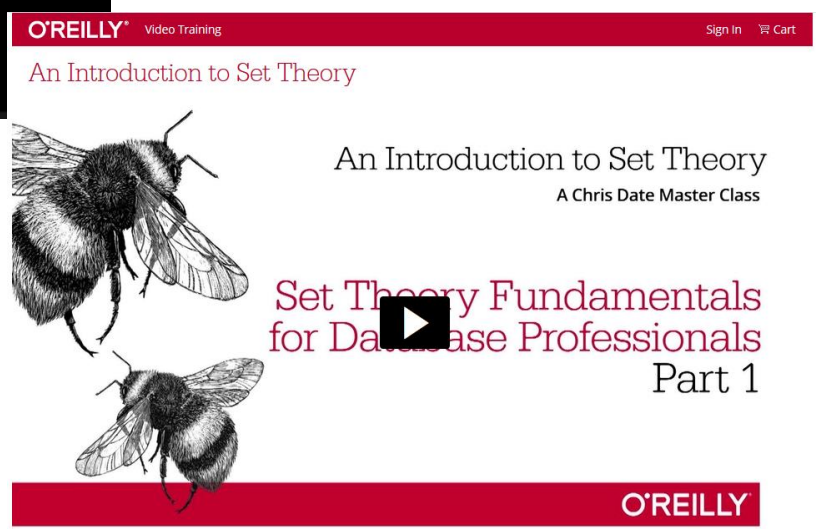


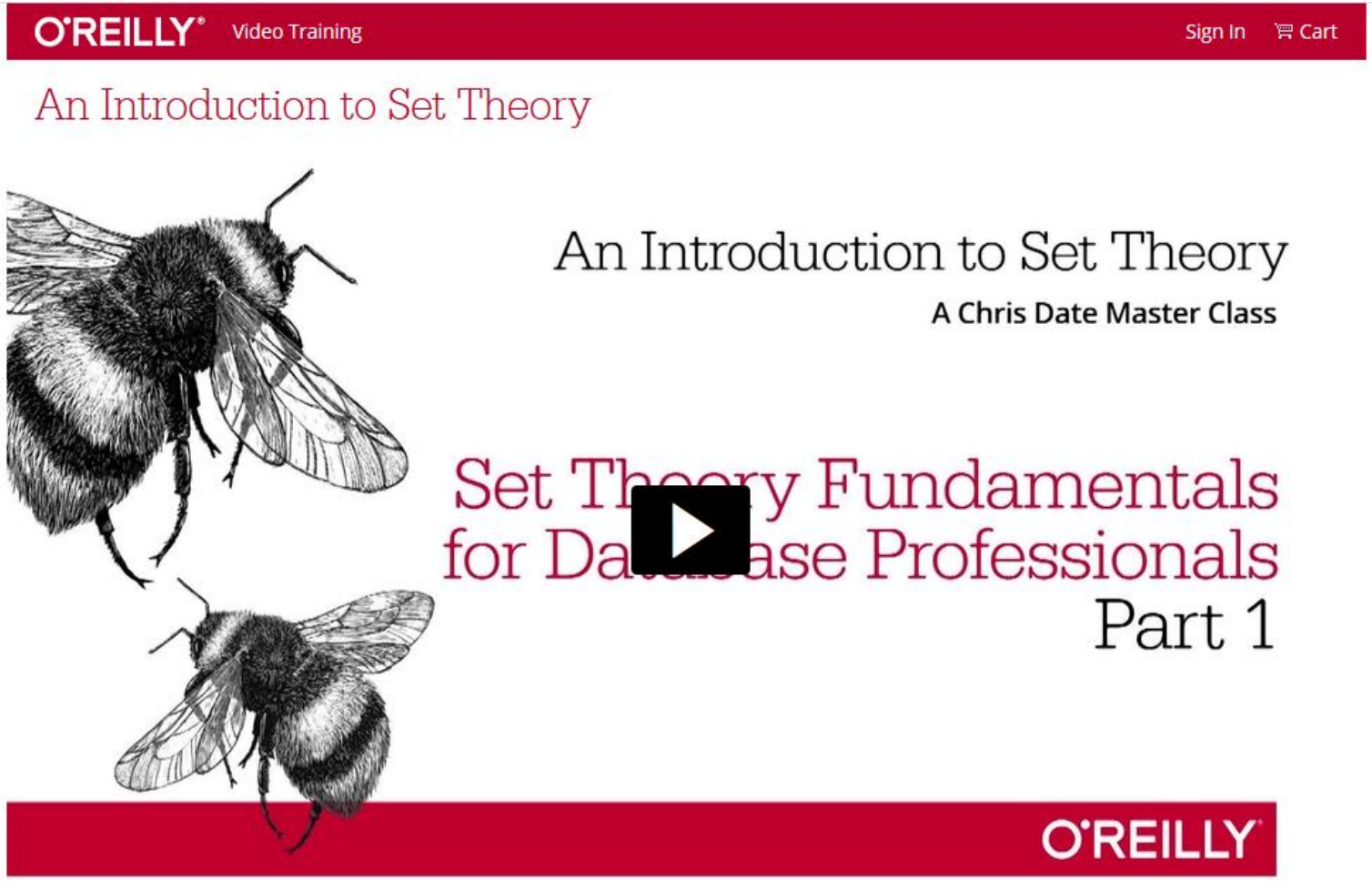


Препоручени курсеви

https://www.pluralsight.com





Препоручене књиге и курсеви

https://www.oreilly.com/